

## **Banza Challenge**

¡Hola! nos alegra que estes participando de este proceso de selección para trabajar en Banza.

En esta oportunidad, para conocer un poco más acerca de tus habilidades, te compartimos un pequeño challenge, el cual será analizado por nuestro equipo.

No te vamos a exigir la identificación de una solución óptima, aunque, en caso de que la encuentres, será reconocido.

La fecha límite de entrega es 1 semana desde la recepción, pero si por la razón que fuera consideras que necesitas un poco más de tiempo, no dudes en avisarnos y extendemos el plazo, no hay problema.

Por último, si llegas a tener alguna consulta, puedes evacuarla a través del siguiente correo:  
Ezequiel Ortiz: [eortiz@ad-cap.com.ar](mailto:eortiz@ad-cap.com.ar)

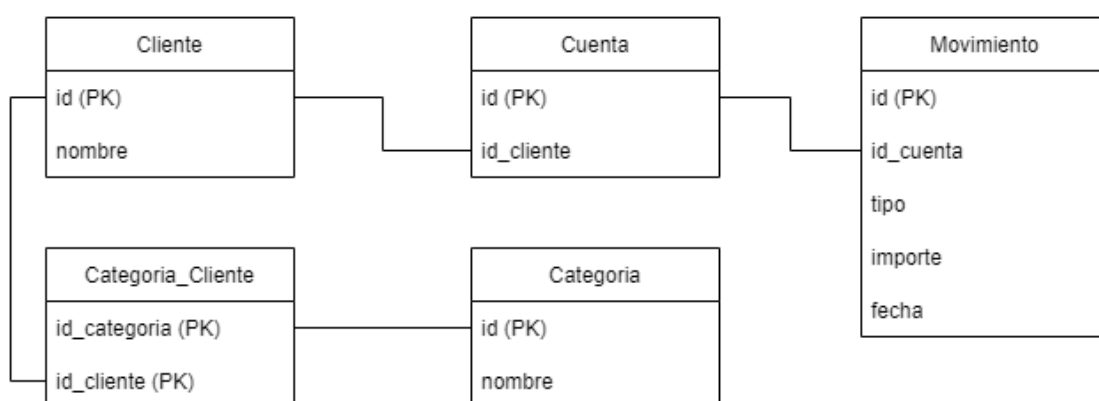
¡¡Muchos éxitos!!

### **Consigna**

Crear una API REST utilizando FastAPI que brinde la siguiente funcionalidad básica de un sistema de control de movimientos monetarios.

#### Requerimientos Técnicos:

- Deberás desarrollar una API REST utilizando FastAPI.
- Los datos mostrados deben ser persistidos en una base de datos relacional.
- La API deberá exponer endpoints que devuelvan datos en JSON.



La misma debe proporcionar los siguientes endpoints:

#### Cientes

- Registrar/Editar un cliente.
- Eliminar un cliente.
- Listar todos los clientes.
- Agregar cliente a categoría.
- Consultar un cliente, sus cuentas y categorías.
- Consultar saldo disponible en cada una de sus cuentas. El saldo se construye utilizando los movimientos

Nota: No es necesario proporcionar vía API REST el ABM de cuentas y/o categorías.

#### Movimientos

- Registrar un movimiento, puede ser del *tipo Ingreso* o *Egreso* de dinero.
- Eliminar un movimiento.
- Consultar un movimiento.

Por otro lado, deben existir el siguiente método:

- La clase Cuenta debe exponer el método **get\_total\_usd** que valore el saldo disponible en dólares utilizando la cotización expuesta en la API de <https://www.dolarsi.com/api/api.php?type=valoresprincipales>, "Dolar Bolsa" y retorne el valor correspondiente.

---

#### Consideraciones finales:

- Tener en cuenta que existen dos tipos diferentes de movimientos. **Ingreso** (impacto positivo en saldo disponible) y **Egreso** (impacto negativo en saldo disponible).
- Al crear un movimiento de tipo Egreso se debe validar que haya suficiente saldo disponible en la cuenta del cliente, en caso de no contar con saldo suficiente se debe retornar un error de validación.
- Implementar Tests para validar los endpoints.
- El código fuente debe ser subido a un repositorio público, el cual debe ser proporcionado para su corrección, se debe incluir también el dump de la base de datos.